Machine Learning I

Bjoern Andres (Lectures)
Shengxian Zhao and Jerome Thiessat (Exercises)

Machine Learning for Computer Vision
TU Dresden

Winter Term 2021/2022

Contents. This part of the course is about structured data, the structured learning problem and the structured inference problem.

Motivation. Even the most general learning and inference problem w.r.t. constrained data $(S, X, x, \mathcal{Y})$ we have considered is too restrictive for certain applications:

- Attributes $x_s \in X$ are defined for single elements $s \in S$ only.
- Dependencies between decisions $y_s, y_{s'} \in \{0, 1\}$ for distinct $s, s' \in S$ are only due to hard constraints definded by the feasible set $\mathcal{Y} \subset \{0, 1\}^S$.

Example: Pixel classification: Given a digital image, we need to decide for every pixel $s \in S$, by the contents of the image around that pixel, whether the pixel is of interest $(y_s = 1)$ or not of interest $(y_s = 0)$.

Typically, decisions at neighboring pixels $s, s' \in S$ are more likely to be equal $(y_s = y_{s'})$ than unequal $(y_s \neq y_{s'})$, and we wish to learn how this increased probability depends on the contents of the image.

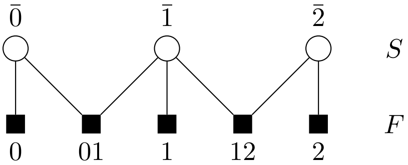
The mathematical abstractions of learning we have considered so far are insufficient to express these dependencies.

In order to lift this restriction, we will define the **supervised structured learning** problem and the **structured inference** problem in which

- attributes are associated with subsets of S
- decisions can be tied by probabilistic dependencies.

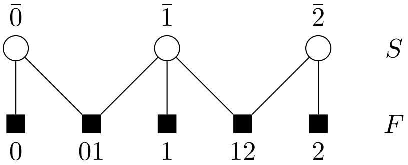
More specifically, we will

- introduce a family $H : \Theta \to \mathbb{R}^{X \times Y}$ of functions that quantify by $H_\theta(x, y)$ how incompatible attributes $x \in X$ are with a combination of decisions $y \in \{0, 1\}^S$
- define supervised structured learning as a problem of finding one function from this family
- define structured inference as the problem of finding a combination of decisions $y \in \{0, 1\}^S$ that minimizes $H_\theta(x, \cdot)$.



Definition. A triple (S, F, E) is called a **factor graph** with **variable nodes** S and **factor nodes** F iff $S \cap F = \emptyset$ and $(S \cup F, E)$ is a bipartite graph such that $\forall e \in E \, \exists s \in S \, \exists f \in F \colon e = \{s, f\}$.

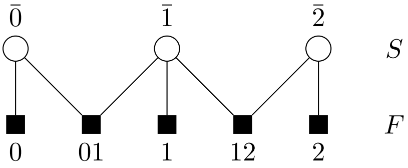
- For any factor node $f \in F$, we denote by $S_f = \{s \in S \mid \{s, f\} \in E\}$ the set of those variable nodes that are neighbors of f.
- For any variable node $s \in S$, we denote by $F_s = \{f \in F \mid \{s, f\} \in E\}$ the set of those factor nodes that are neighbors of s.



Definition. A tuple $T = (S, F, E, \{X_f\}_{f \in F}, x)$ is called **unlabeled structured data** iff the following conditions hold:

- (S, F, E) is a factor graph
- Every set X_f is non-empty, called the **attribute space** of f
- $x \in \prod_{f \in F} X_f$, where the Cartesian product $\prod_{f \in F} X_f$ is called the **attribute space** of T.

A tuple $(S, F, E, \{X_f\}_{f \in F}, x, y)$ is called **labeled structured data** iff $(S, F, E, \{X_f\}_{f \in F}, x)$ is unlabeled structured data, and $y \in \{0, 1\}^S$.

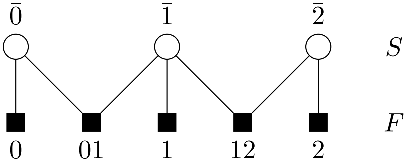


Definition. W.r.t. any labeled structured data $(S, F, E, \{X_f\}_{f \in F}, x, y)$,

- the attribute space $X = \prod_{f \in F} X_f$
- the set $Y = \{0, 1\}^S$
- any $\Theta \neq \emptyset$ and family of functions $H : \Theta \to \mathbb{R}^{X \times Y}$
- any $R : \Theta \to \mathbb{R}_0^+$, called a **regularizer**
- any $L : \mathbb{R}^Y \times Y \to \mathbb{R}_0^+$, called a **loss function**
- any $\lambda \in \mathbb{R}_0^+$, called a **regularization parameter**,

the instance of the **supervised structured learning problem** has the form

$$\inf_{\theta \in \Theta} \quad \lambda R(\theta) + L(H_\theta(x, \cdot), y) \qquad (1)$$



Definition. With respect to

- any unlabeled structured data $T = (S, F, E, \{X_f\}_{f \in F}, x)$
- any $\hat{H} \colon X \times \{0,1\}^S \to \mathbb{R}$

the instance of the **structured inference problem** has the form

$$\min_{y \in \{0,1\}^S} \quad \hat{H}(x, y) \tag{2}$$

Summary.

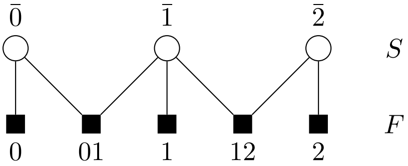
- **Structured data** consists of a factor graph (S, F, E) and attributes $x_f \in X_f$ for every factor $f \in F$.
- The **structured learning problem** is an optimization problem whose feasible solutions θ define functions $H_\theta : X \times Y \to \mathbb{R}$ whose values $H_\theta(x, y)$ quantify an incompatibility of attributes $x \in X$ and combinations of decisions $y \in \{0, 1\}^S$.
- The **structured inference problem** consists in finding decisions $y \in \{0, 1\}^S$ compatible with given attributes $x \in X$, by minimizing a given incompatibility function $\hat{H}(x, \cdot)$.

Contents. This part of the course is about supervised structured learning of conditional graphical models.

Definition. For any factor graph $G = (S, F, E)$, a function $H : \{0,1\}^S \to \mathbb{R}$ is said to **factorize** w.r.t. G iff, for every $f \in F$, there exists a function a function $h_f : \{0,1\}^{S_f} \to \mathbb{R}$, called a **factor** of H, such that

$$\forall y \in \{0,1\}^S: \quad H(y) = \sum_{f \in F} h_f(y_{S_f}) \ . \tag{3}$$

Example: A function $H : \{0,1\}^S \to \mathbb{R}$ factorizes w.r.t. the factor graph



iff there exist suitable functions $h_0, h_{01}, h_1, h_{12}, h_2$ such that, for any $y \in \{0,1\}^S$: $H(y) = h_0(y_{\bar{0}}) + h_1(y_{\bar{1}}) + h_2(y_{\bar{2}}) + h_{01}(y_{\bar{0}}, y_{\bar{1}}) + h_{12}(y_{\bar{1}}, y_{\bar{2}})$.

Definition. A tuple $(S, F, E, \{X_f\}_{f\in F}, \Theta, \{h_f\}_{f\in F})$ is called a **conditional graphical model** with attribute space $X := \prod_{f\in F} X_f$ and parameter space Θ iff the following conditions hold:

- (S, F, E) is a factor graph
- $\Theta \neq \emptyset$
- For every $f \in F$:
  - X_f is non-empty, called the **attribute space** of f
  - $h_f : \Theta \to \mathbb{R}^{X_f \times \{0,1\}^{S_f}}$, called a **factor**.

The family $H : \Theta \to \mathbb{R}^{X \times \{0,1\}^S}$ such that

$$\forall \theta \in \Theta \; \forall x \in X \; \forall y \in \{0,1\}^S : \quad H_\theta(x, y) = \sum_{f\in F} h_{f\theta}(x_f, y_{S_f}) \tag{4}$$

is called the family of **energy functions** of the conditional graphical model.

Family of Functions

- We consider a conditional graphical model $(S, F, E, \{X_f\}_{f \in F}, \Theta, \{h_f\}_{f \in F})$ and its family H of energy functions.
- We assume that Θ is a finite-dimensional, real vector space, i.e., there exists a finite, non-empty set J and $\Theta = \mathbb{R}^J$.
- We assume that every function h_f is linear in Θ, i.e., for every $f \in F$, there exists a $\varphi_f : X_f \times \{0,1\}^{S_f} \to \mathbb{R}^J$ such that for any $x_f \in X_f$, any $y_{S_f} \in \{0,1\}^{S_f}$ and any $\theta \in \Theta$:

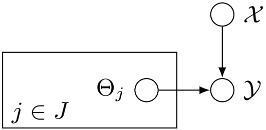
$$h_{f\theta}(x_f, y_{S_f}) = \langle \theta, \varphi_f(x_f, y_{S_f}) \rangle \tag{5}$$

For convenience, we define $\xi : X \times \{0,1\}^S \to \mathbb{R}^J$ such that for any $x \in X$ and any $y \in \{0,1\}^S$:

$$\xi(x,y) = \sum_{f \in F} \varphi_f(x_f, y_{S_f}) \tag{6}$$

Thus, we obtain for any $\theta \in \Theta$, any $x \in X$ and any $y \in Y$:

$$\begin{aligned} H_\theta(x,y) &= \sum_{f \in F} h_{f\theta}(x_f, y_{S_f}) \\ &= \sum_{f \in F} \langle \theta, \varphi_f(x_f, y_{S_f}) \rangle \\ &= \left\langle \theta, \sum_{f \in F} \varphi_f(x_f, y_{S_f}) \right\rangle \\ &= \langle \theta, \xi(x,y) \rangle \end{aligned} \tag{7}$$



Probabilistic Model

- Let $\mathcal{X}$ be a random variable whose value is an element $x \in X$ of the attribute space.
- Let $\mathcal{Y}$ be a random variable whose value is a combination of decisions $y \in \{0,1\}^S$
- For any $j \in J$, let Θ_j a random variable whose value is a parameter $\theta_j \in \mathbb{R}$

Factorization

- We assume:

$$P(\mathcal{X}, \mathcal{Y}, \Theta) = P(\mathcal{Y} \mid \mathcal{X}, \Theta)\, P(\mathcal{X}) \prod_{j \in J} P(\Theta_j) \tag{8}$$

- Thus:

$$\begin{aligned} P(\Theta \mid \mathcal{X}, \mathcal{Y}) &= \frac{P(\mathcal{X}, \mathcal{Y}, \Theta)}{P(\mathcal{X}, \mathcal{Y})} \\ &= \frac{P(\mathcal{Y} \mid \mathcal{X}, \Theta)\, P(\mathcal{X})\, \prod_{j \in J} P(\Theta_j)}{P(\mathcal{X}, \mathcal{Y})} \\ &\propto P(\mathcal{Y} \mid \mathcal{X}, \Theta) \prod_{j \in J} P(\Theta_j) \end{aligned} \tag{9}$$

Distributions

Definition. For any conditional graphical model, the **partition function** $Z\colon X \times \Theta \to \mathbb{R}$ and **Gibbs distribution** $p\colon X \times \{0,1\}^S \times \Theta \to [0,1]$ are defined by the forms

$$Z(x,\theta) = \sum_{y \in \{0,1\}^S} e^{-H_\theta(x,y)} \tag{10}$$

$$p(y,x,\theta) = \frac{1}{Z(x,\theta)} e^{-H_\theta(x,y)} \tag{11}$$

We consider a $\sigma \in \mathbb{R}^+$ and

$$p_{\mathcal{Y}|\mathcal{X},\Theta}(y,x,\theta) = \frac{1}{Z(x,\theta)} e^{-H_\theta(x,y)} \tag{12}$$

$$\forall j \in J\colon \qquad p_{\Theta_j}(\theta_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\theta_j^2/2\sigma^2} \quad . \tag{13}$$

Lemma. Estimating maximally probable parameters θ, given attributes x and decisions y, i.e.,

$$\underset{\theta \in \mathbb{R}^J}{\operatorname{argmax}} \quad p_{\Theta|\mathcal{X},\mathcal{Y}}(\theta, x, y)$$

is identical to the supervised structured learning problem w.r.t. L, R and λ such that

$$\begin{aligned}
L(H_\theta(x,\cdot), y) &= H_\theta(x,y) + \ln Z(x,\theta) && (14)\\
&= H_\theta(x,y) + \ln \sum_{y' \in \{0,1\}^S} e^{-H_\theta(x,y')} && (15)\\
&= \langle \theta, \xi(x,y) \rangle + \ln \sum_{y' \in \{0,1\}^S} e^{-\langle \theta, \xi(x,y') \rangle} && (16)\\
R(\theta) &= \|\theta\|_2^2 && (17)\\
\lambda &= \frac{1}{2\sigma^2} && (18)
\end{aligned}$$

Lemma: The first and second partial derivatives of the logarithm of the partition function have the forms

$$\frac{\partial}{\partial \theta_j} \ln Z = \frac{1}{Z(x,\theta)} \sum_{y' \in \{0,1\}^S} (-\xi_j(x,y')) e^{-\langle \theta, \xi(x,y') \rangle} \tag{19}$$

$$= \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(-\xi_j(x,y')) \tag{20}$$

$$\frac{\partial^2}{\partial \theta_j \, \partial \theta_k} \ln Z = \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x,y') \xi_k(x,y')) - \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x,y')) \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_k(x,y'))$$

$$= \mathsf{COV}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x,y'), \xi_k(x,y')) \tag{21}$$

Lemma: Supervised structured learning of a conditional graphical model is a convex optimization problem.

Lemma: Estimating maximally probable decisions y, given attributes x and parameters θ, i.e.

$$\underset{y \in \{0,1\}^S}{\operatorname{argmax}} \quad p_{\mathcal{Y}|\mathcal{X},\Theta}(x, y, \theta) \tag{22}$$

is identical to the structured inference problem with $\hat{H}(x, y) = H_\theta(x, y)$.

Summary. Supervised structured learning of conditional graphical models whose factors are linear functions is a convex optimization problem.

Conditional Graphical Models II

Contents. This part of the course introduces algorithms for supervised structured learning of conditional graphical models.

On the one hand, supervised structured learning of conditional graphical models whose factors are linear functions is a **convex** optimization problem.

Thus, it can be solved exactly by means of the **steepest descent algorithm** with a tolerance parameter $\epsilon \in \mathbb{R}_0^+$:

$\theta := 0$
repeat
  $d := \nabla_\theta L(H_\theta(x, \cdot), y)$
  $\eta := \operatorname{argmin}_{\eta' \in \mathbb{R}} L(H_{\theta - \eta' d}(x, \cdot), y)$ (line search)
  $\theta := \theta - \eta d$
  if $\|d\| < \epsilon$
    return θ

On the other hand, computing the gradient naïvely takes time $O(2^{|S|})$:

$$
\begin{aligned}
-\frac{\partial}{\partial \theta_j}\ln Z &= \mathbb{E}_{y'\sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x,y')) \\
&= \frac{1}{Z(x,\theta)}\sum_{y'\in\{0,1\}^S} \xi_j(x,y')\, e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \frac{1}{Z(x,\theta)}\sum_{y'\in\{0,1\}^S}\sum_{f\in F} \varphi_{fj}(x_f,y'_{S_f})\, e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \frac{1}{Z(x,\theta)}\sum_{f\in F}\sum_{y'_{S(f)}\in\{0,1\}^{S(f)}}\sum_{y'_{S\setminus S(f)}\in\{0,1\}^{S\setminus S(f)}} \varphi_{fj}(x_f,y'_{S(f)})\, e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \sum_{f\in F}\sum_{y'_{S(f)}\in\{0,1\}^{S(f)}} \varphi_{fj}(x_f,y'_{S(f)})\frac{1}{Z(x,\theta)}\sum_{y'_{S\setminus S(f)}\in\{0,1\}^{S\setminus S(f)}} e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \sum_{f\in F}\sum_{y'_{S(f)}\in\{0,1\}^{S(f)}} \varphi_{fj}(x_f,y'_{S(f)})\, p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)}\mid x,\theta) \\
&= \sum_{f\in F}\mathbb{E}_{y'_{S(f)}\sim p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}}(\varphi_{fj}(x_f,y'_{S(f)}))
\end{aligned}
$$

Computing the gradient requires that we compute

- the partition function

$$Z(x,\theta) = \sum_{y' \in \{0,1\}^S} e^{-\langle \theta, \xi(x,y') \rangle} \tag{23}$$

- for every factor $f \in F$, the so-called **factor marginal**

$$p_{\mathcal{Y}_{S(f)} \mid \mathcal{X},\Theta}(y'_{S(f)} \mid x, \theta) = \frac{1}{Z(x,\theta)} \sum_{y'_{S \setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} e^{-\langle \theta, \xi(x,y') \rangle} \tag{24}$$

- for every factor $f \in F$, the expectation value

$$\sum_{y'_{S(f)} \in \{0,1\}^{S(f)}} \varphi_{fj}(x_f, y'_{S(f)})\, p_{\mathcal{Y}_{S(f)} \mid \mathcal{X},\Theta}(y'_{S(f)} \mid x, \theta) \ . \tag{25}$$

The challenge is to sum the function

$$\psi_\theta(x, y') := e^{-\langle \theta, \xi(x, y') \rangle} \tag{26}$$

over assignments of 0 or 1 to linearly many (24) or all (23) variables y'.

Defining

$$\psi_{f\theta}(x_f, y'_{S(f)}) = e^{-\langle \theta, \varphi_f(x_f, y'_{S(f)}) \rangle} \tag{27}$$

we obtain

$$\begin{aligned}
\psi_\theta(x, y') &= e^{-\langle \theta, \xi(x, y') \rangle} \\
&= e^{-\sum_{f \in F} \langle \theta, \varphi_f(x_f, y_{S(f)}) \rangle} && (28) \\
&= \prod_{f \in F} e^{-\langle \theta, \varphi_f(x_f, y_{S(f)}) \rangle} && (29) \\
&= \prod_{f \in F} \psi_{f\theta}(x_f, y_{S(f)}) \ . && (30)
\end{aligned}$$

Thus, the challenge in (24) and (23) is to compute a sum of a product of functions. Specifically:

$$Z(x,\theta) = \sum_{y' \in \{0,1\}^S} \prod_{f \in F} \psi_{f\theta}(x_f, y_{S(f)}) \tag{31}$$

$$p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)} \mid x, \theta) = \frac{1}{Z(x,\theta)} \sum_{y'_{S\setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} \prod_{f \in F} \psi_{f\theta}(x_f, y_{S(f)}) \tag{32}$$

- One approach to tackle this problem is to sum over variables recursively.
- In order to avoid redundant computation, Kschischang et al. (2001) define partial sums.

Definition (Kschischang et al. (2001)) For any variable node $s \in S$ and any factor node $f \in F$, the functions

$$m_{s \to f}, m_{f \to s} \colon \{0, 1\} \to \mathbb{R} \ , \tag{33}$$

called **messages**, are defined such that for all $y_s \in \{0, 1\}$:

$$m_{s \to f}(y_s) = \prod_{f' \in F(s) \setminus \{f\}} m_{f' \to s}(y_s) \tag{34}$$

$$m_{f \to s}(y_s) = \sum_{y_{S(f) \setminus \{s\}}} \psi_{f\theta}(x_f, y_{S(f)}) \prod_{s' \in S(f) \setminus \{s\}} m_{s' \to f}(y_{s'}) \tag{35}$$

Lemma. If the factor graph is acyclic, messages are defined recursively by (34) and (35), beginning with the messages from leaves. Moreover, for any $s \in S$ and any $f \in F$:

$$Z(x, \theta) = \sum_{y_s \in \{0,1\}} \prod_{f' \in F(s)} m_{f' \to s}(y_s) \tag{36}$$

$$p_{\mathcal{Y}_{S(f)} | \mathcal{X}, \Theta}(y'_{S(f)} \mid x, \theta) = \frac{1}{Z(x, \theta)} \psi_{f\theta}(x_f, y_{S(f)}) \prod_{s' \in S(f)} m_{s' \to f}(y_{s'}) \tag{37}$$

The recursive computation of messages is known as **message passing**.

Summary

- For conditional graphical models whose factor graph is **acylic**, the supervised structured learning problem can be solved efficiently by means of the steepest descent algorithm and message passing.
- For conditional graphical models whose factor graph is **cyclic**, the definition of messages is cyclic as well. The partition function and marginals cannot be computed by message passing in general.
- A heuristic without guarantee of correctness or even convergence is to initialize all messages as normalized constant functions and to update messages according to some schedule, e.g., synchronously. This heuristic is commonly known as **loopy belief propagation**.

Conditional Graphical Models III

Contents. This part of the course introduces algorithms for supervised structured inference with conditional graphical models.

The **inference problem** w.r.t. a **conditional graphical model** has the form of an unconstrained binary optimization problem:

$$\underset{y \in \{0,1\}^S}{\operatorname{argmin}} \; H_\theta(x, y) \tag{38}$$

It is NP-hard. (This can be shown, e.g., by reduction of binary integer programming, which is one of Karp's 21 problems).

We consider transformations that change one decision at a time:

Definition. For any $s \in S$, let $\mathrm{flip}_s \colon \{0,1\}^S \to \{0,1\}^S$ such that for any $y \in \{0,1\}^S$ and any $t \in S$:

$$\mathrm{flip}_s[y](t) = \begin{cases} 1 - y_t & \text{if } t = s \\ y_t & \text{otherwise} \end{cases} . \tag{39}$$

The greedy local search algorithm w.r.t these transformations is known as **Iterated Conditional Modes**, or ICM (Besag 1986).

$y' = \mathsf{icm}(y)$
choose $s \in \underset{s' \in S}{\operatorname{argmin}} \; H_\theta(x, \mathrm{flip}_{s'}[y]) - H_\theta(x, y)$
if $H_\theta(x, \mathrm{flip}_s[y]) < H_\theta(x, y)$
$\quad y' := \mathsf{icm}(\mathrm{flip}_s[y])$
else
$\quad y' := y$

- The **inference problem** consists in computing the minimum of a sum of functions:

$$
\begin{aligned}
& \underset{y \in \{0,1\}^S}{\operatorname{argmin}} \; H_\theta(x, y) \\
= & \underset{y \in \{0,1\}^S}{\operatorname{argmin}} \; \sum_{f \in F} h_{f\theta}(x_f, y_{S(f)})
\end{aligned}
\tag{40}
$$

- This problem is analogous to that of computing the sum of a product of functions (from the previous lecture) in that both $(\mathbb{R}, \min, +)$ and $(\mathbb{R}, +, \cdot)$ are commutative semi-rings.
- This analogy is sufficient to transfer the idea of **message passing**, albeit with messages adapted to the $(\mathbb{R}, \min, +)$ semi-ring:

Definition. (Kschischang 2001) For any variable node $s \in S$ and any factor node $f \in F$, the functions

$$\mu_{s \to f}, \mu_{f \to s} \colon \{0,1\} \to \mathbb{R} \ , \tag{41}$$

called **messages**, are defined such that for all $y_s \in \{0,1\}$:

$$\mu_{s \to f}(y_s) = \sum_{f' \in F(s) \setminus \{f\}} \mu_{f' \to s}(y_s) \tag{42}$$

$$\mu_{f \to s}(y_s) = \min_{y_{S(f) \setminus \{s\}}} \psi_{f\theta}(x_f, y_{S(f)}) + \sum_{s' \in S(f) \setminus \{s\}} \mu_{s' \to f}(y_{s'}) \tag{43}$$

Lemma. If the factor graph is acyclic, messages are defined recursively by (42) and (43), beginning with the messages from leaves. Moreover, for any $s \in S$:

$$\begin{aligned}
&\operatorname*{argmin}_{y \in \{0,1\}^S} \; H_\theta(x, y) \\
=& \min_{y \in \{0,1\}^S} \; \sum_{f \in F} h_{f\theta}(x_f, y_{S(f)}) \\
=& \min_{y_s \in \{0,1\}} \sum_{f' \in F(s)} \mu_{f' \to s}(y_s)
\end{aligned} \tag{44}$$

Proof. Analogous to that of Lemma 18 in the lecture notes.

Summary

- For conditional graphical models whose factor graph is **acylic**, the inference problem can be solved efficiently by means of **min-sum message passing**.
- For conditional graphical models whose factor graph is **cyclic**, one local search algorithm for the inference problem is known as **Iterated Conditional Modes (ICM)**.